Contribution to T1 Standards Project

STANDARDS PROJECT: Coding and Performance Specifications for Multimedia Communications on Internet Services (T1A1-15).

TITLE: Low Bandwidth Techniques for Estimating Temporal Delays Between Input and Output Video Sequences

SOURCE: NTIA/ITS

CONTACT:

Margaret Pinson
Voice: (303) 497-3579
Fax: (303) 497-5323
e-mail: margaret@its.bldrdoc.gov

Stephen Wolf
Voice: (303) 497-3771
Fax: (303) 497-5323
e-mail: steve@its.bldrdoc.gov

DATE: May, 1999

DISTRIBUTION: Working Group T1A1.5

KEY WORDS: video delay, temporal alignment, low bandwidth features

ABSTRACT: This contribution presents a new technique for estimating temporal delays between input and output video streams from video teleconferencing systems. The discussion in this contribution is limited to "constant alignment," defined as an alignment process that computes the same fixed alignment offset, or delay, for every output video frame in a given output video sequence. Constant alignment can (1) serve as a starting point for variable alignment techniques, where each output frame can have a unique alignment offset, or delay, and (2) be used to align low frame rate output video sequences in preparation for measuring video quality parameters. The constant alignment technique that is presented is applicable for real-time in-service monitoring since it utilizes a set of computationally efficient low bandwidth features that are extracted from the input and output video streams.

Low Bandwidth Techniques for Estimating Temporal Delays Between Input and Output Video Sequences

1. Introduction

Section 6.4.1 of ANSI standard T1.801.03-1996 [1] describes an algorithm that computes a constant alignment[1] offset, or delay, between an input and output video stream. In brief, the constant alignment algorithm in ANSI T1.801.03 computes a temporal information (TI) feature[2] for each video frame, forms input and output sequences of these input and output features, normalizes these input and output feature sequences by the standard deviation of that sequence, computes the variance of the difference between the normalized input and output feature sequences for each alignment shift under consideration, and chooses the alignment that minimizes the variance of this difference. From an intuitive standpoint, this process produces an alignment offset where changes in output scene motion match changes in input scene motion. The algorithm can be used for real-time in-service measurements since only one number for each video frame is required to be transmitted between the input and output ends of the video system.

When applied to video systems that transmit a high number of frames per second, the measurement is reliable, accurate, and robust. As the number of frames per second decreases, the temporal information of the output video stream becomes discontinuous due to the presence of repeated frames (i.e., no motion). Figure 1 demonstrates this condition for a very low frame rate video system, hypothetical reference circuit (HRC) 20 [2], and a high motion scene, ftball [3]. Areas of repeated frames (i.e., no or low motion) and frame updates (i.e., motion spikes) are clearly observable. As the transmission frame rate decreases from 30 frames per second, the input and output motion sequences become increasingly dissimilar and this produces a flatter correlation function, which in turn produces a more unreliable estimate of the overall time shift. This is demonstrated by the right hand graph in Figure 1 which is a plot of the correlation function for HRC 20 and the scene ftball. If the input and output motion curves were identical, the correlation function would dip to zero at the correct alignment. As these two curves become more dissimilar, the dip in the correlation function becomes less pronounced as a smaller portion of the output variance is canceled by the input when aligned. Therefore, the magnitude of the correlation function at best alignment can provide a measure of the reliability of the alignment offset.

Output video sequences that contain field/frame repeating or variable delay do not have a single correct constant alignment. When a constant alignment is desirable for these cases, any of a set of reasonable alignments might be chosen. Constant alignment can serve as the starting point for variable alignment algorithms like the one given in ANSI T1.801.04-1997 [4] and can be used to align low frame rate output video sequences in preparation for measuring video quality parameters. This contribution outlines an improved constant alignment algorithm which computes a reasonable alignment for video sequences containing field/frame repeats or variable delays. This new constant alignment algorithm produces delay estimates that are identical to the ANSI T1.801.03 constant alignment algorithm when applied to output video sequences with no field/frame repeating or variable delays. When field/frame repeating or variable

[1] The term "constant alignment" as used in this contribution and ANSI T1.801.03 refers to a temporal alignment process that computes the same fixed alignment offset, or delay, for every output video frame in a given output video sequence. This is contrasted with "variable alignment" techniques mentioned in section 6.4.2 of ANSI T1.801.03 and described in ANSI T1.801.04-1997 where each output video frame can have a unique alignment offset, or delay.

[2] The temporal information (TI) feature measures the motion energy and is computed as the root mean square (rms) of the difference between successive luminance frames.

delays are present in the output, the new algorithm produces improved estimates of constant alignment by utilizing additional low bandwidth features extracted from the input and output video streams.

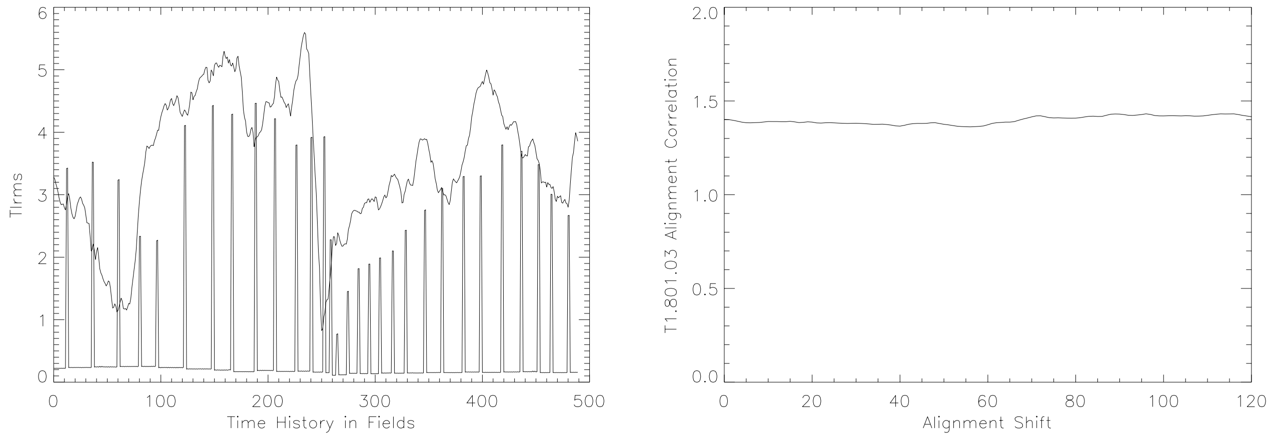

Figure 1. Scene “football” HRC 20; Left - ANSI TI features, Right - ANSI TI correlation function.

2. Description of Features

The improved constant alignment algorithm uses four new low bandwidth features in succession, trying to achieve alignment at each stage before proceeding to the next feature. The first feature is a field-based implementation of the frame-based TI feature used in section 6.4.1 of ANSI T1.801.03. This TI feature uses temporal field differences spaced 2 fields apart in time (i.e., 1 frame). Two other TI features are used that are based on temporal field differences spaced 4 fields apart (2 frames) and 10 fields apart (5 frames). One feature is used that is based on the mean of the field. The same normalization and correlation techniques used in ANSI T1.801.03 are also used on each of these new feature sequences. The magnitude of the correlation function for a particular feature’s best alignment offset indicates the reliability of that feature’s alignment. Each of the four new features characterize the video differently. Thus, each works best for different amounts of frame repeating and varying delay; and one may succeed where another one fails.

Before alignment, the output video sequence must be calibrated. Luminance gain and offsets are removed, spatial shifts are removed, and the active video area is determined. See [5] for a description of these calibration techniques. All further computations are made on the active portion of the calibrated video only. In the following feature definitions, the luminance field is noted as Y, and the time n when this field occurs is denoted as t_n. Pixels of Y are further subscripted by row and column, i and j, respectively, so that an individual pixel is denoted as $Y(i,j,t_n)$.

2.1 TI2 Feature: Two Field Difference Temporal Information

The first feature used will be denoted as TI2 and is calculated as shown in Figure 2. The TI2 feature is essentially the same feature as that used by the ANSI T1.801.03 constant alignment algorithm, but computed on fields instead of frames.

To compute TI2 at time t_n, consider field $Y(t_n)$ and the previous field of the same type, $Y(t_{n-2})$, and compute

$$TI2(i,j,t_n) = Y(i,j,t_n) - Y(i,j,t_{n-2})$$

for each pixel, and then compute

$$TI2(t_n) = rms_{space}[TI2(i,j,t_n)],$$

where rms_{space} is the root mean square function over space. We have found that if the standard deviation of the time history ($stdev_{time}$) of $TI2(t_n)$ is 0.05 or less, then this feature does not contain enough variance for alignment to be reliably calculated. In other words, alignment results using $TI2(t_n)$ become unreliable when

$$stdev_{time}\,[TI2(t_n)] \leq TI\_THRESHOLD,$$

where

$$TI\_THRESHOLD = 0.05.$$

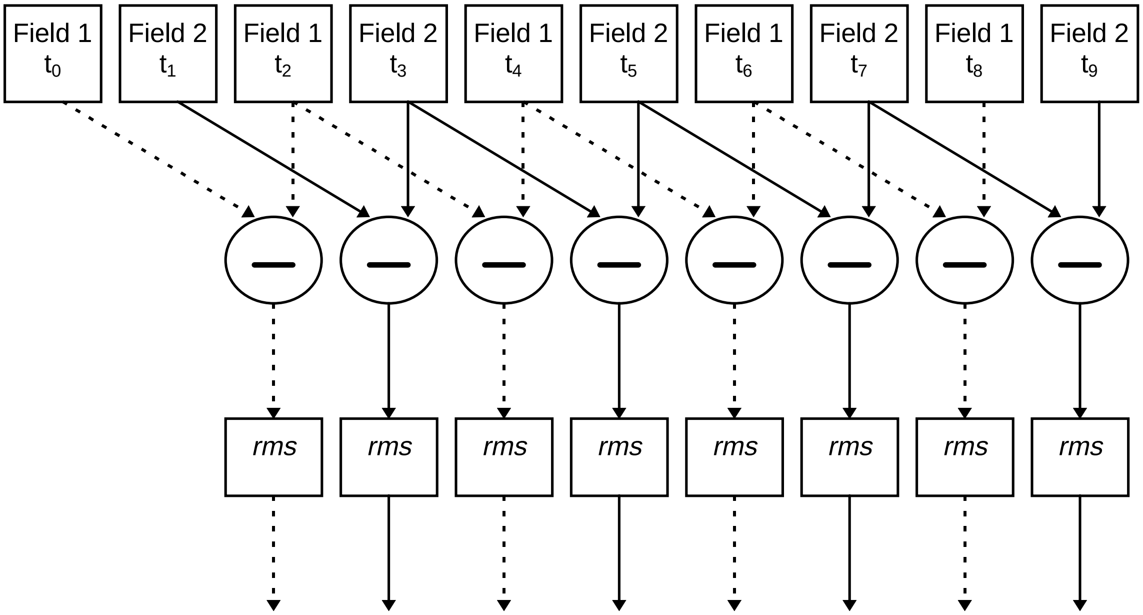

Figure 2. Diagram depicting the method of calculating $TI2(t_n)$.

2.2 TI4 Feature: Four Field Difference Temporal Information

The second feature used will be denoted as TI4 and is calculated as shown in Figure 3. The TI4 feature is based on a temporal difference spaced four fields apart (i.e., two frames apart). This feature smoothes the temporal information using a wider filter than TI2 and eliminates frame repeats in the TI waveform for systems that have one or fewer consecutive frame repeats.

To compute TI4 at time t_n, consider field $Y(t_n)$ and the field of the same type two frames ago, $Y(t_{n-4})$, and compute

$$TI4(i,j,t_n) = Y(i,j,t_n) - Y(i,j,t_{n-4})$$

for each pixel, and then compute

$$TI4(t_n) = rms_{space}[TI4(i,j,t_n)].$$

As was the case with $TI2(t_n)$, if the standard deviation of the time history of $TI4(t_n)$ is 0.05 or less, then this feature does not contain enough variance for alignment to be reliably calculated.

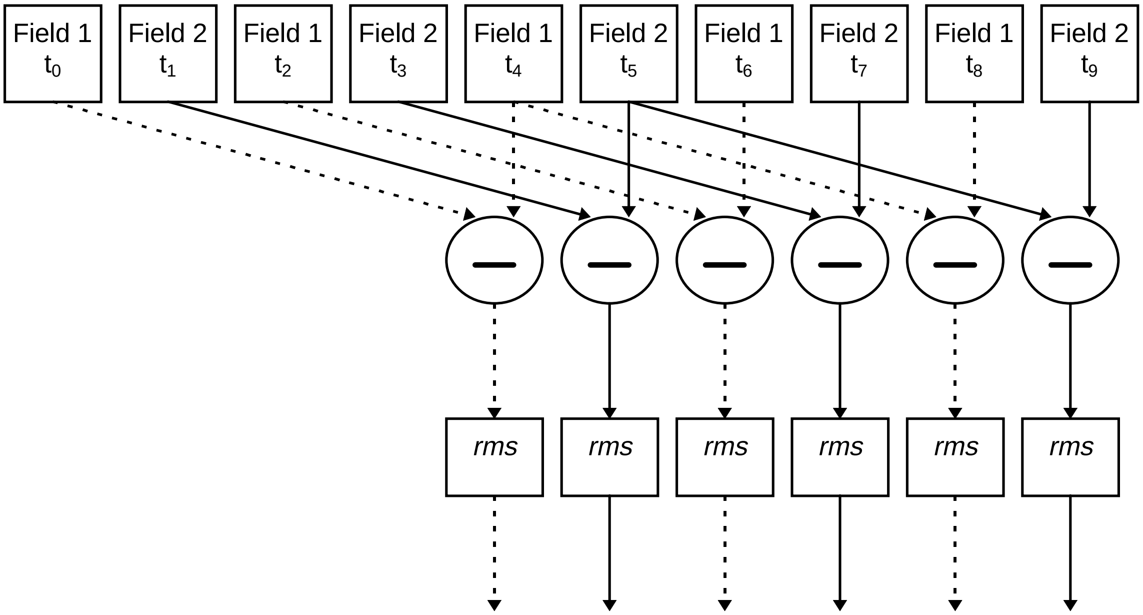

Figure 3. Diagram depicting the method of calculating TI4(t_n).

2.3 Ymean Feature: Average Luminance Level

The third feature used will be denoted as Ymean and is calculated as the average luminance level of a field. To compute Ymean at time t_n, consider field Y(t_n) and compute

$$\text{Ymean}(t_n) = \textit{mean}_{\text{space}}[\text{Y}(i,j,t_n)],$$

where *mean*$_{\text{space}}$ is the mean function over space. If the standard deviation of the time history of Ymean(t_n) is 0.5 or less, then this feature does not contain enough variance for alignment to be reliably calculated. In other words, alignment results using Ymean(t_n) become unreliable when

$$\textit{stdev}_{\text{time}}\ [\text{Ymean}(t_n)] \leq \text{Y\_THRESHOLD},$$

where

$$\text{Y\_THRESHOLD} = 0.5.$$

2.4 TI10 Feature: Ten Field Difference Temporal Information

The fourth feature used will be denoted as TI10. The TI10 feature is based on a temporal difference spaced ten fields apart (i.e., five frames apart). This feature smoothes the temporal information using a wider filter than TI4 and eliminates frame repeats in the TI waveform for systems that have four or fewer consecutive frame repeats.

To compute TI10 at time t_n, consider field Y(t_n) and the field of the same type five frames ago, Y(t_{n-10}), and compute

$$\text{TI10}(i,j,t_n) = \text{Y}(i,j,t_n) - \text{Y}(i,j,t_{n-10})$$

for each pixel, and then compute

$$\mathrm{TI10}(t_n) = rms_{\mathrm{space}}[\mathrm{TI10}(i,j,t_n)].$$

As was the case with $\mathrm{TI2}(t_n)$ and $\mathrm{TI4}(t_n)$, if the standard deviation of the time history of $\mathrm{TI10}(t_n)$ is 0.05 or less, then this feature does not contain enough variance for alignment to be reliably calculated.

3. Feature Sequence Correlation

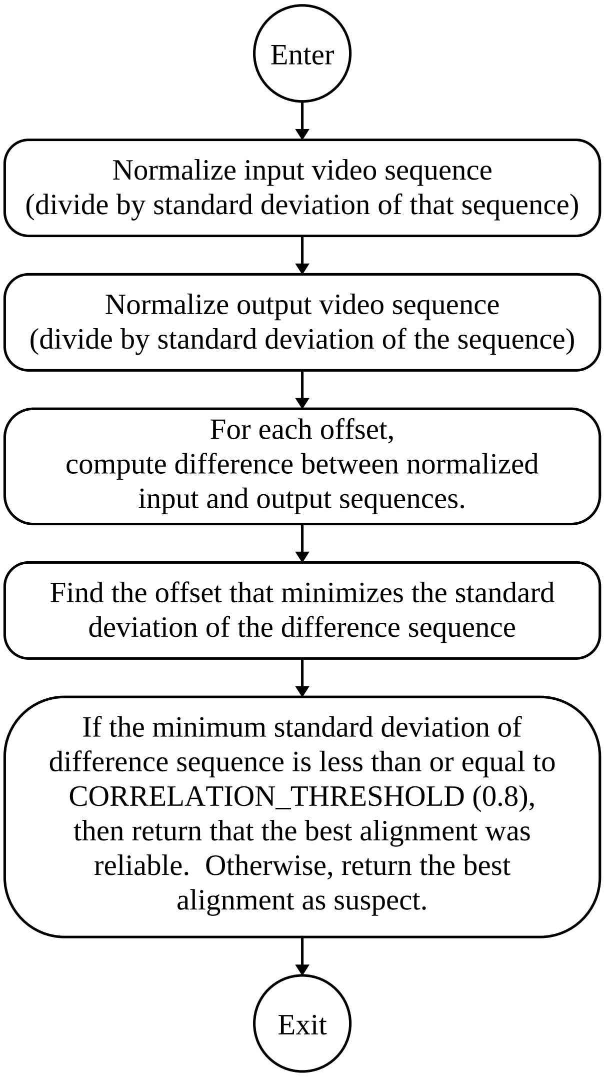
Enter
Normalize input video sequence (divide by standard deviation of that sequence)
Normalize output video sequence (divide by standard deviation of the sequence)
For each offset, compute difference between normalized input and output sequences.
Find the offset that minimizes the standard deviation of the difference sequence
If the minimum standard deviation of difference sequence is less than or equal to CORRELATION_THRESHOLD (0.8), then return that the best alignment was reliable. Otherwise, return the best alignment as suspect.
Exit

Figure 4. Correlation Algorithm Description.

Given a sequence of output video features, $\{a_{out}(t_0), a_{out}(t_1), a_{out}(t_2), \ldots\ a_{out}(t_{m-1})\}$; normalize (divide) each element in the sequence by the standard deviation of that sequence.

For each alignment guess, g, in the range of alignments to be considered ($g \leq 0$), compute a sequence of input video features, $\{\ a_{in}(t_{0+g}), a_{in}(t_{1+g}), a_{in}(t_{2+g}), \ldots\ a_{in}(t_{m-1+g})\}$; normalize (divide) each element in the sequence by the standard deviation of that sequence.

Take the resulting normalized output and input sequences and compute the difference between those sequences,

$D(g) = \{a_{in}(t_{0+g})\text{-}a_{out}(t_0), a_{in}(t_{1+g})\text{-}a_{out}(t_1), a_{in}(t_{2+g})\text{-}a_{out}(t_2), \ldots, a_{in}(t_{m-1+g})\text{-}a_{out}(t_{m-1})\}$.

Compute the standard deviation over time of the difference sequence for each offset g, namely

$S(g) = stdev_{\mathrm{time}}(D(g))$.

The minimum S(g) (denoted S_{min}), and its offset g_{min} is the best alignment indicated by this feature. If

$S_{min} \leq \mathrm{CORRELATION\_THRESHOLD}$,

where

CORRELATION_THRESHOLD = 0.8,

then the correlation between the selected output sequence and the input video stream is probably reliable. Otherwise, the results of this correlation are suspect (the output feature waveform is too dissimilar to the input feature waveform for a reliable estimate of constant alignment).

4. Entire Algorithm

The following describes how to apply the four features of section 2 and the correlation algorithm of section 3 to achieve the final estimate of the alignment offset g.

1. Apply the correlation algorithm from section 3 to TI2 from section 2.1. If the results are reliable, accept that alignment offset.

2. If the results from step 1 are suspect, apply the correlation algorithm from section 3 to TI4 from section 2.2. If the results are reliable, accept that alignment offset.
3. If the results from step 2 are suspect, apply the correlation algorithm from section 3 to Ymean from section 2.3. If the results are reliable, accept that alignment offset.
4. If the results from step 3 are suspect, apply the correlation algorithm from section 3 to TI10 from section 2.4. If the results are reliable, accept that alignment offset.
5. If the results from step 4 are suspect, then report that the video sequences cannot be aligned.

5. Threshold Optimization

The behavior of the constant alignment algorithm of section 4 is influenced by the setting of three empirical thresholds: TI_THRESHOLD, Y_THRESHOLD, and CORRELATION_THRESHOLD. TI_THRESHOLD sets the amount of variability, or information, that must be present in the TI feature streams before alignment is attempted. When the standard deviation over time of the TI2, TI4, and TI10 feature streams fall below TI_THRESHOLD (recommended value of 0.05), then we have found that the TI feature streams do not contain enough variance for alignment to be reliably calculated. This could be due to lack of scene motion (i.e., a still scene), or perhaps a constant amount of scene motion (e.g., the amount of motion present for any given field in the sequence is non-zero but constant). Y_THRESHOLD sets the amount of variability, or information, that must be present in the Ymean feature stream before alignment is attempted. When the standard deviation over time of the Ymean feature stream falls below Y_THRESHOLD (recommended value of 0.5), then the Ymean feature stream does not contain enough variance for alignment to be reliably calculated. This could be due to lack of scene motion or perhaps a constant brightness level (e.g., the average brightness level for any given field in the sequence is constant).

If the TI2 and TI4 features fail because of constant scene motion or low frame rate, the Ymean feature can succeed if the scene's brightness level changes. If TI2, TI4, and Ymean all fail, then a final attempt is made to perform alignment using the TI10 feature, which further enhances the scene motion and covers over the periods of very low frame rate. Raising the TI_THRESHOLD will cause the Ymean feature to be used more often. In turn, raising the Y_THRESHOLD will cause the TI10 feature to be used more often. If the TI and Y thresholds are raised too high, there is an increased likelihood that the algorithm will reach step 5 and report that the video sequences cannot be aligned.

The feature sequence correlation function given in section 3 uses a CORRELATION_THRESHOLD with a recommended value of 0.8. If the input and output feature sequences were identical, then their variances would cancel at correct alignment (i.e., S_{min} would be 0.0). On the other hand, if the input and output waveforms were statistically independent, then their variances would add (i.e., S_{min} would be equal to the square root of 2.0, since the input and output are each normalized to have a variance of 1.0). A CORRELATION_THRESHOLD of 0.8 imposes the constraint that at least 36% of the normalized output variance must be canceled by the input (i.e., $1.0 - [0.8]^2$) in order to declare a good alignment. Raising the correlation threshold will increase the likelihood of falsely concluding that an alignment step has succeeded, when in fact it has failed. Lowering the correlation threshold will favor later steps in the alignment algorithm and increase the likelihood of reaching step 5 (i.e., the video sequences cannot be aligned).

Table 1 summarizes the recommended values for the three thresholds used by the alignment algorithm.

Table 1 – Recommended Values for Thresholds Used by Alignment Algorithm

Threshold	Recommended Value
TI_THRESHOLD	0.05
Y_THRESHOLD	0.5
CORRELATION_THRESHOLD	0.8

6. Example Results

This constant alignment algorithm was tested on video scenes and HRCs from the T1A1 video test data set [2]. Due to limitations with our disk storage for sampled images, the algorithm was tested on one third of the T1A1 data set. Of the chosen set, three fourths was selected at random while one fourth was selected due to its relatively poor objective to subjective correlation results (lack of a good temporal alignment is one possible cause of poor objective to subjective correlation results). All of these clips in the chosen data set were subsequently aligned by visual examination, noting a middle reasonable alignment (a visual compromise), the earliest reasonable alignment, and the latest reasonable alignment. The constant alignment algorithm's results all fell within the range of reasonable alignments chosen visually and usually fell near the middle of that range. The example plots that follow are taken from two scenes (5row1, vtc1nw) and four HRCs (2 - VHS, 11 - QCIF 128 kb/sec, 18 - CIF 168 kb/sec, 24 - CIF 1536 kb/sec). The scene 5row1 depicts five people sitting in a row, conversing at a wooden table while the scene vtc1nw is a close-up shot of a woman's head and shoulders in front of a gray background [3].

Figure 5 shows a plot of the input and output feature TI2 for the scene 5row1 and HRC 2. This clip was successfully aligned on the first step of the algorithm.

Figure 6 and Figure 7 depict the TI2 and the Ymean features for the scene 5row1 and HRC 11, respectively. This clip has an average frame rate of about 10 frames per second. The feature TI2 depicted in Figure 6 was unable to reliably align the clip. The TI4 feature (not depicted) was also unable to align this particular clip. However, the clip was successfully aligned on step 3 of the algorithm using the Ymean feature depicted in Figure 7.

Figure 8 and Figure 9 depict the TI2 and TI4 features for the scene vtc1nw and HRC 11, respectively. This clip has a frame rate of 15 frames per second. The TI2 feature depicted in Figure 8 is unable to reliably align this clip. However, the clip was successfully aligned on step 2 of the algorithm using the TI4 feature depicted in Figure 9.

The final example depicts the alignment of scene vtc1nw and HRC 18. The feature TI2 is plotted in Figure 10. Note that the frame repeat rate of this clip decreases during periods of high motion, such as toward the end of the clip. The feature TI2 probably fails to reliably align the clip due to the areas of low frame rate during the high motion periods. The feature TI4 (see Figure 11) has a similar problem. Because the average luminance level of the clip is fairly constant over time, the Ymean feature (see Figure 12) does not contain reliable alignment information either. However, the clip was successfully aligned on step 4 of the algorithm using the TI10 feature, which smoothes over the periods of very low frame rate (see Figure 13).

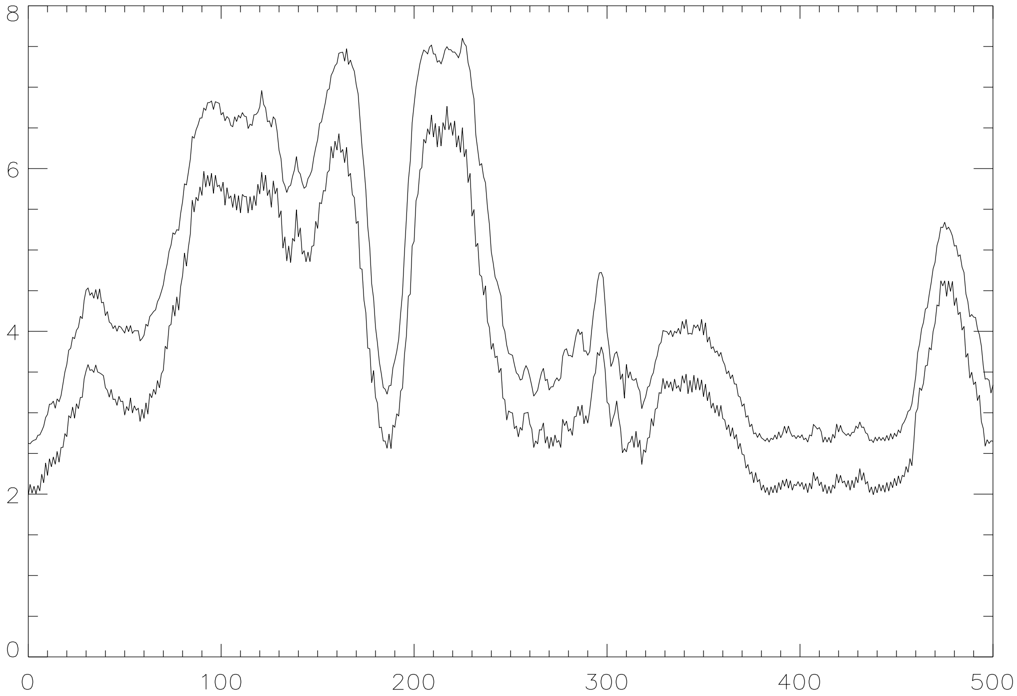

Figure 5. Aligned input and output feature TI2 for the scene 5row1 and HRC 2.

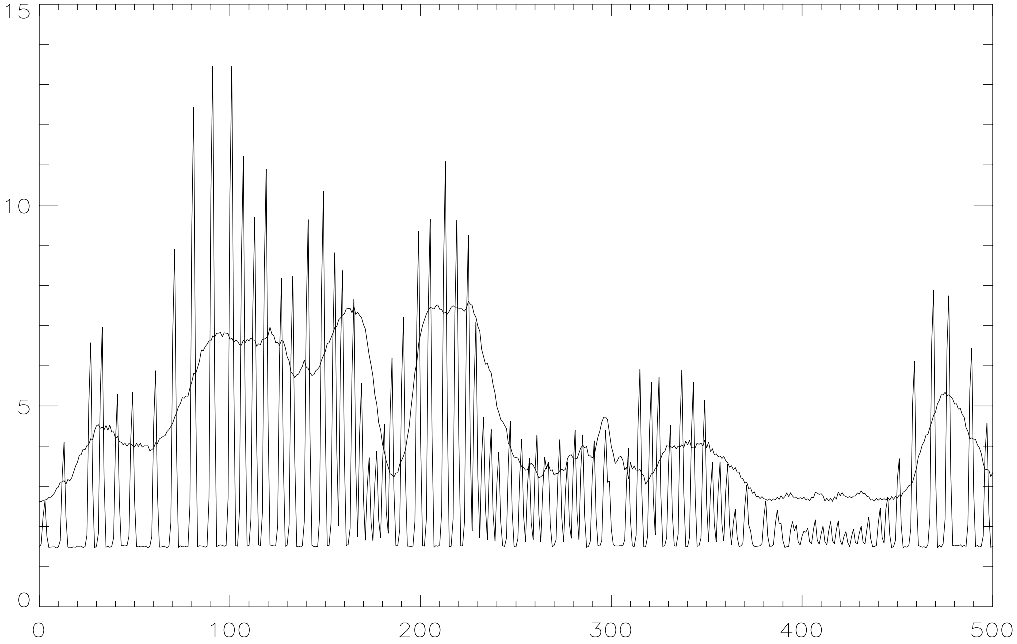

Figure 6. Input and output feature TI2 for the scene 5row1 and HRC 11, failed alignment.

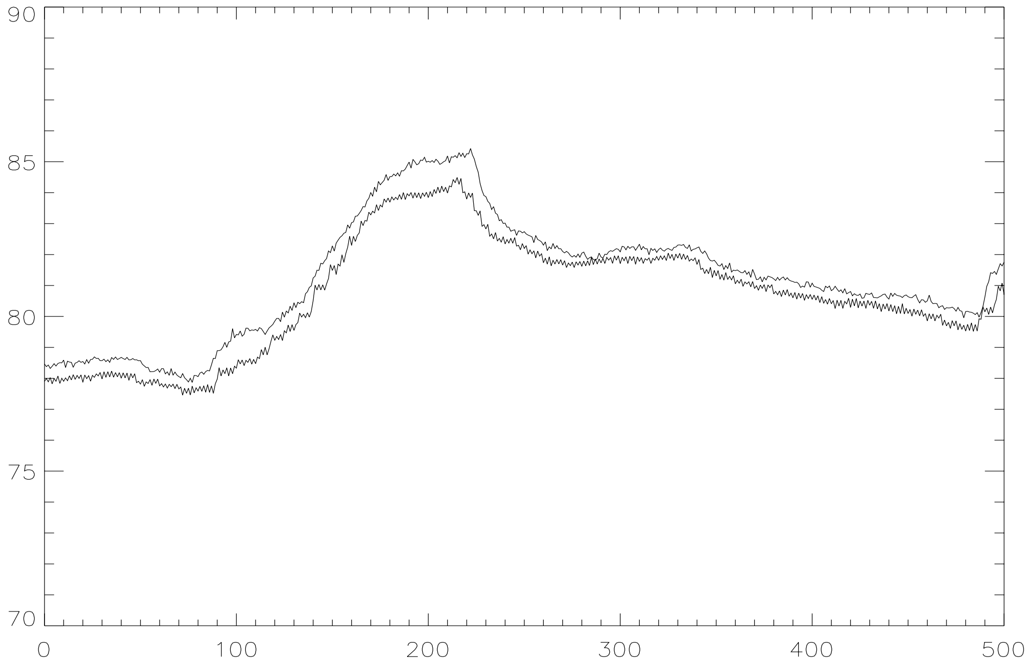

Figure 7. Aligned input and output feature Ymean for the scene 5row1 and HRC 11.

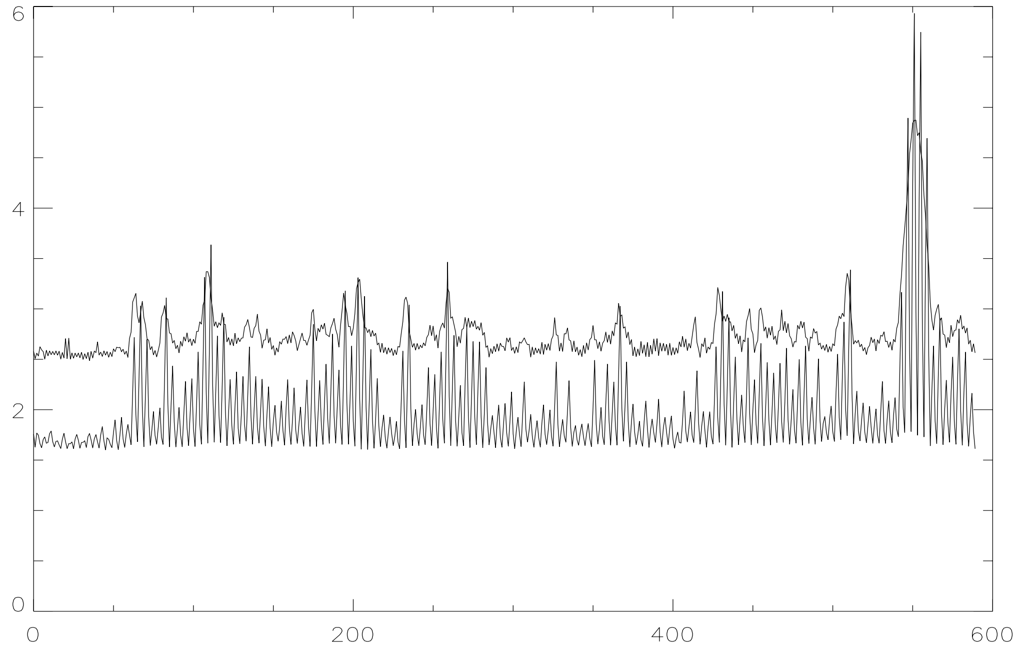

Figure 8. Input and output feature TI2 for the scene vtc1nw and HRC 24, failed alignment.

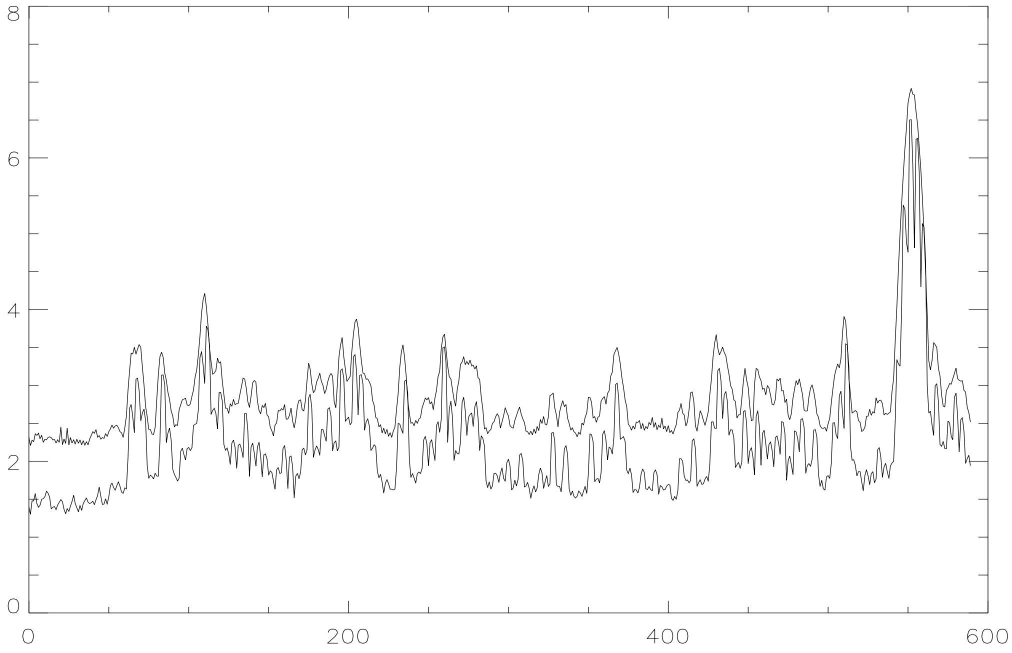

Figure 9. Aligned input and output feature TI4 for the scene vtc1nw and HRC 24.

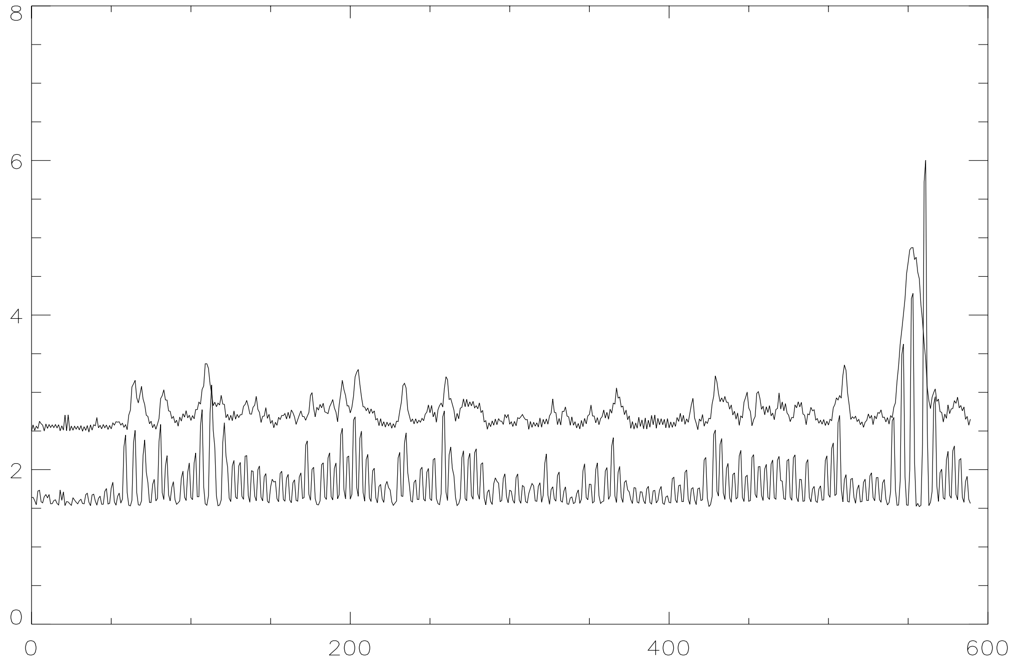

Figure 10. Input and output feature TI2 for the scene vtc1nw and HRC 18, failed alignment.

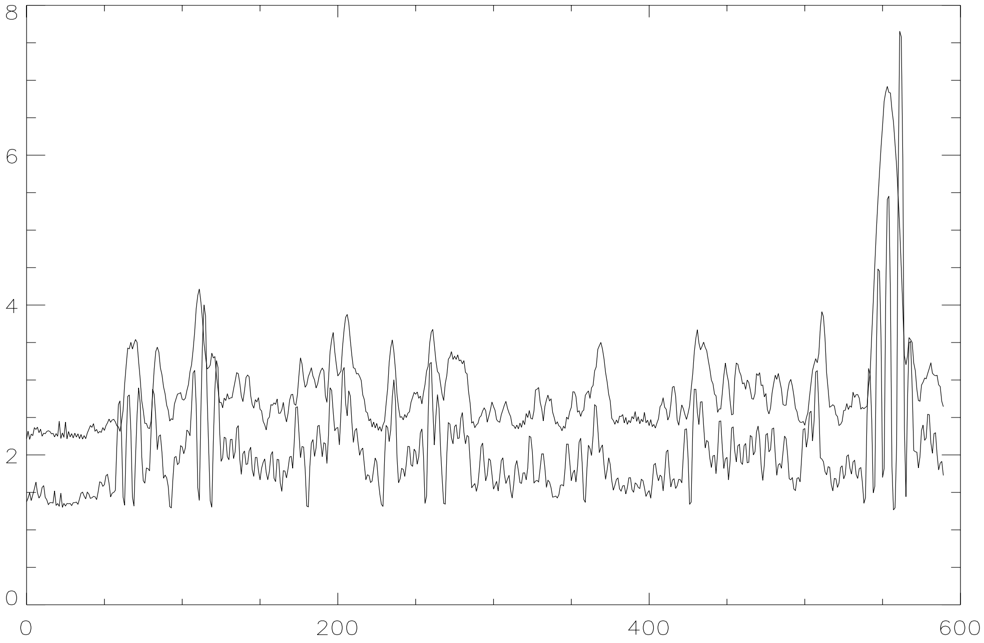

Figure 11. Input and output feature TI4 for the scene vtc1nw and HRC 18, failed alignment.

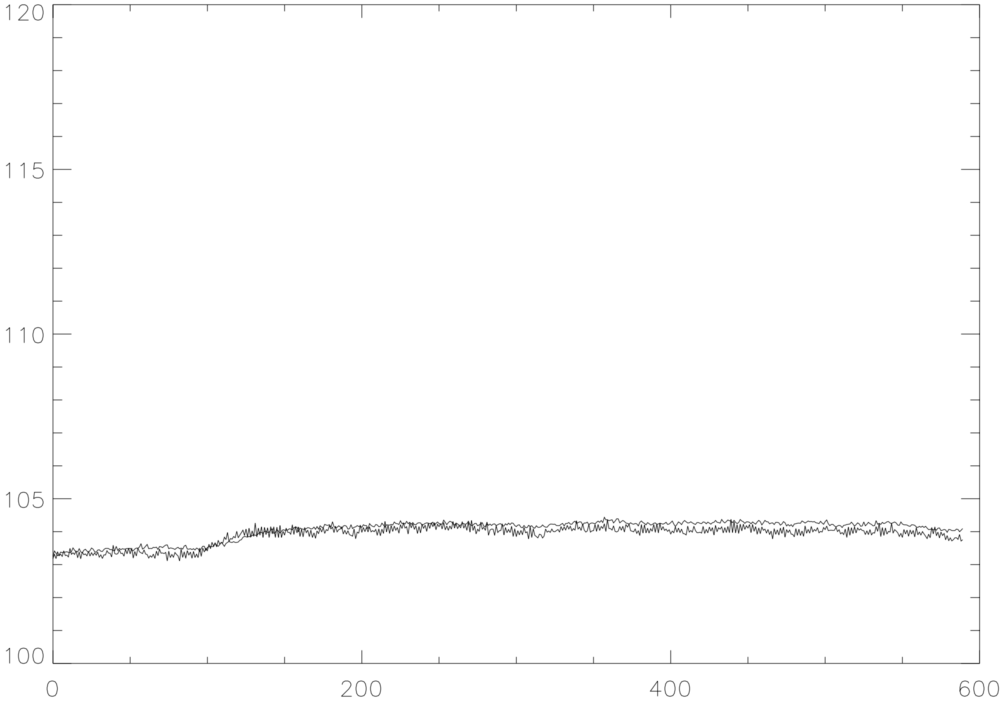

Figure 12. Input and output feature Ymean for the scene vtc1nw and HRC 18, failed alignment.

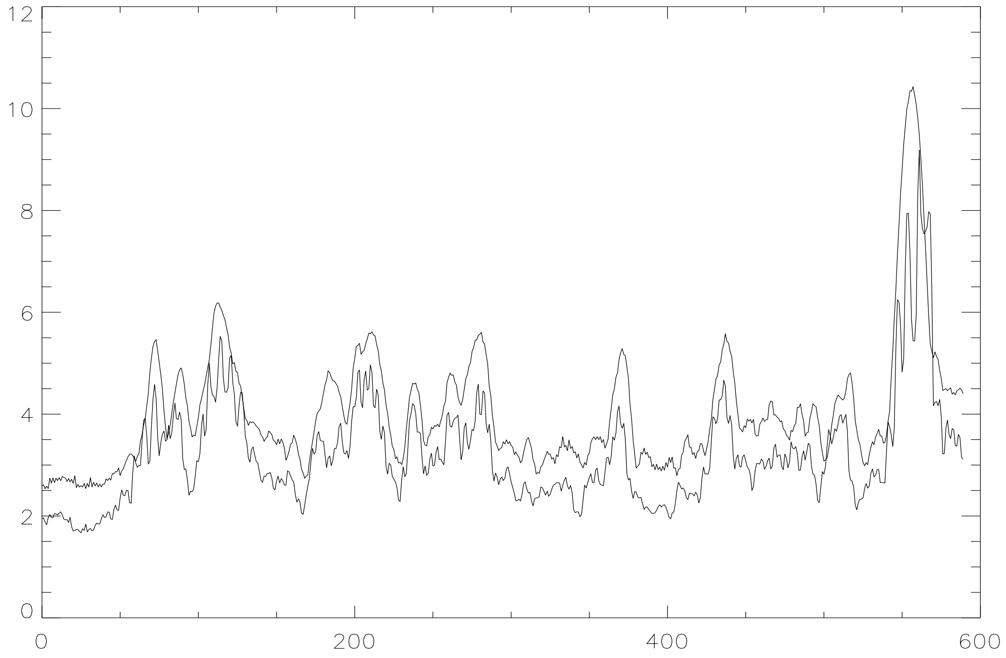

Figure 13. Aligned input and output feature TI10 for the scene vtc1nw and HRC 18.

7. Observations and Conclusions

The constant alignment algorithm presented in ANSI T1.801.03 may be improved upon by using additional low bandwidth field-based features. The four new features, TI2, TI4, Ymean, and TI10, appear to provide reasonable constant alignment for all of the T1A1 data subset that was examined (approximately one third of the total number of video clips). Other widths of TI were tried, such as a six field wide TI, eight field wide TI, twelve field wide TI, etc. However, these additional TI features do not appear to provide any additional improvements. The average luminance level, Ymean, has proven to be useful in estimating constant alignment, particularly when the TI features fail due to low frame rates.

As an interesting note, the alignments preferred by the TI features and the Ymean feature differ. The TI features tend to match low motion portions of the input and output features, so that relatively still portions of the scene will appear best aligned when variable delays are present. Ymean tends to match high motion portions of the input and output video, since these portions of the video sequence tend to contain the greatest luminance level variance. Thus, high-motion portions of the scene will appear best aligned when clips with variable video delays are aligned using the Ymean feature.

An alternative strategy would have been to compute TI with an increasing width in frames, ten frames, twenty frames, thirty frames, etc., rather than switching over to the luminance level feature Ymean. Although that technique would probably work reasonably well, it would require the calculation of many more features since the largest TI width would have to be greater than the largest number of times a frame could be repeated. In the T1A1 data set, this time extent is greater than two seconds. Also, whenever the TI width is larger than the smallest TI width necessary to align a clip, then useful temporal alignment information may be lost, and this may decrease the accuracy of the constant alignment. Replacing Ymean with a set of increasing width TI features would also increase computation time and algorithmic complexity. Hence, the Ymean feature is a more economical approach. While the alignments Ymean produces are slightly less accurate than those from TI2 and TI4, Ymean has the advantage that it is not significantly effected by frame repeats.

The fourth feature, TI10, is primarily included to cover the cases where TI2, TI4, and Ymean all fail: clips with small changes in average luminance level (possibly due to extremely small amounts of motion) and low output frame rates. In these cases, use of a wider TI filter (e.g., TI10) amplifies and smoothes the limited scene motion to the point where alignment may become feasible. In the T1A1 data set, several video teleconferencing scenes fell into this category.

8. References

[1] ANSI T1.801.03-1996, "American National Standard for Telecommunications – Digital Transport of One-Way Video Signals – Parameters for Objective Performance Assessment," American National Standards Institute, 11 West 42nd Street, New York, New York 10036.

[2] A.C. Morton, "Subjective Test Plan (Tenth and Final Draft)," ANSI T1A1 Contribution T1A1.5/94-118R1, October 3, 1993.

[3] ANSI T1.801.01-1995, "American National Standard for Telecommunications – Digital Transport of Video Teleconferencing / Video Telephony Signals – Video Test Scenes for Subjective and Objective Performance Assessment," American National Standards Institute, 11 West 42nd Street, New York, New York 10036.

[4] ANSI T1.801.04-1997, "American National Standard for Telecommunications – Multimedia Communications Delay, Synchronization, and Frame Rate Measurement," American National Standards Institute, 11 West 42nd Street, New York, New York 10036.

[5] S. Wolf et al., "Objective and Subjective Measures of MPEG Video Quality," ANSI T1A1 Contribution T1A1.5/96-121, October 28, 1996